

LIST OF CURRENT CLAIMS

1. - 11. (Cancelled)

12. (Currently Amended) ~~An operand memory stack for use in a~~ A calculating machine containing an operand memory stack and a processing unit for processing individual operands according to a program, the operand memory stack comprising:

operands of different lengths that are contiguously packed as a stack; and

a type memory comprising memory elements of constant length that store, for each operand stored in the operand memory stack, a type information that is coded and which contains information about the length of a corresponding operand;

wherein the length of a particular operand type is stored in a table in dependence on the corresponding coded type information.

13. (Currently Amended) ~~An operand memory stack~~ A calculating machine according to claim 12, wherein the type memory is formed separate from the operand memory stack as a stack with constant length stack elements.

14. (Currently Amended) ~~An operand memory stack~~ A calculating machine according to claim 12, wherein the type memory is integrated into the operand memory stack such that each operand is directly contiguous to the corresponding coded type information.

15. (Currently Amended) ~~An operand memory stack~~ A calculating machine according to claim 12, wherein the operand memory stack is formed as a virtual stack for a virtual calculating machine.

16. (Previously Presented) ~~An operand memory stack~~ A calculating machine according to claims 12, 13, 14 or 15, further comprising an operand type checking device which is activated at each read access to the operand memory stack.

17. Cancelled

18. (Previously Presented) An operand memory stack according to claims 12, 13 or 14, wherein the operand memory stack is contained within a smart card having an integrated virtual calculating machine.

19. (Previously presented) A method for operating an operand memory stack in a calculating machine, the method comprising:

providing stack elements of the operand memory stack that are used for storing operands of different length in a contiguously packed arrangement;

creating a type memory element of uniform length for each operand in the operand memory stack;

storing coded type information that contains length information about the length of each corresponding operand in the type memory elements;

evaluating said length information at each access to the operand memory stack;
and

storing the length of a particular operand type in a table in dependence on the corresponding coded type information.

20. (Previously Presented) A method according to claim 19, wherein the type memory elements are created in the form of a separate stack.

21. (Previously Presented) A method according to claim 19, wherein the type memory elements are stored contiguously with the corresponding operand memory stack element.

22. (Previously Presented) A method according to claims 19, 20 or 21, wherein a type check is performed at each read access to the operand memory stack.